IN THE CLAIMS

Please amend claims 1-61 to the following:

1. (Previously Amended) A method comprising:

   receiving a first system management interrupt (SMI) with a first and a second processor;

   handling the first SMI with the first processor;

   generating a wake-up signal with the first processor after receiving the first SMI, wherein the wake-up signal references a first memory address of a default SMI handler;

   receiving the wake-up signal with the second processor;

   awakening the second processor, based on the wake-up signal from the first processor; and

   handling the first SMI with the second processor.

2. (Previously amended) The method of claim 1, wherein the first and second processors are logical processors, and wherein the first logical processor includes a first thread and the second logical processor includes a second thread.

3. (Original) The method of claim 1, wherein the first and second processors are physical processors.

4. (Previously Amended) The method of claim 1, wherein handling the first SMI with a first processor comprises: executing the default SMI handler located at the first memory address.

5. (Previously Amended) The method of claim 4, wherein the wake-up signal is a startup inter processor interrupt (SIPI) signal.

6. (Original) The method of claim 5, wherein the first memory address is aligned.

7. (Original) The method of claim 6, wherein the first memory address is 4k aligned.

8. (Previously Amended) The method of claim 5, wherein handling the first SMI with the second processor comprises executing the default SMI handler with the second processor.

9. (Original) The method of claim 8, wherein handling the first SMI with the second processor further comprises patching an instruction pointer to a second memory address.

10. (Original) The method of claim 9, wherein the second memory address is a non-aligned address.

11. (Previously Amended) The method of claim 10, further comprising: executing code at the second memory address after handling the first SMI with the second processor.

12. (Previously Amended) A method comprising:

receiving a first system management interrupt (SMI);

executing code at a first memory location with a first processor in response to the first SMI;

generating a wake-up signal with the first processor;

awakening a second processor, based on a wake-up signal from the first processor; and

executing the code from the first memory location with the second processor, in response to the first SMI after awakening the second processor.

13. (Previously Amended) The method of claim 12, wherein the wake-up signal is based on the first memory location.

14. (Previously Amended) The method of claim 13, wherein the code at the first memory location is default SMI handling code.

15. (Previously Amended) The method of claim 14, wherein the first memory location is located in conventional memory.

16. (Previously Amended) The method of claim 15, wherein the first memory location is aligned.

17. (Original) The method of claim 12, wherein both the first and second processors are logical processors located on the same die.

18. (Original) The method of claim 12, wherein the first and second processors are physical processors located on separate packages.

19. (Previously Amended) The method of claim 12, further comprising

patching an instruction pointer for the second processor to a second memory location.

20. (Previously Amended) The method of claim 19, wherein the second memory location is a non-aligned location.

21. (Previously Amended) The method of claim 20, further comprising:

executing code at the second memory location after patching the instruction pointer to the second memory location.

22. (Previously Amended) The method of claim 12, further comprising generating the SMI before receiving the first SMI.

23. (Original) The method of claim 22, wherein generating the SMI comprises changing the logic level of a pin coupled to a controller hub.

24. (Original) The method of claim 22, wherein an APIC is used to generate the SMI.

25. (Original) The method of claim 24, wherein the APIC is located in the first processor.

26. (Cancelled)

27. (Cancelled)

28. (Cancelled)

29. (Cancelled)

30. (Cancelled)

31. (Cancelled)

32. (Original) A method comprising:

executing system management interrupt (SMI) code with a first processor to handle a SMI for the first processor;

checking if the SMI is a software generated SMI; and

executing the SMI code to handle the SMI for a second processor, if the SMI is software generated.

33. (Original) The method of claim 32, wherein the first processor executes the SMI code to handle the SMI for the second processor, if the SMI is software generated.

34. (Original) The method of claim 32, wherein the second processor executes the SMI code to handle the SMI for the second processor, if the SMI is software generated.

35. (Original) The method of claim 32, wherein the first processor has a first system management base (SMBase) address.

36. (Previously Amended) The method of claim 35, wherein the second processor has a second SMBase address.

37. (Original) The method of claim 36, wherein said SMI code is located at first memory location, which has an offset from the first SMBase address.

38. (Previously Amended) The method of claim 37, wherein executing said SMI code to handle the SMI for the second processor comprises:

changing a target SMBase from the first SMBase to the second SMBase; and

executing the SMI code using the second SMBase as the target SMBase.

39. (Original) The method of claim 38, further comprising returning the target SMBase of the SMI handler to the first SMBase after executing the SMI code to handle the SMI for the second processor.

40. (Previously Amended) An apparatus comprising:

a controller to generate a first system management interrupt (SMI);

a first logical processor, coupled to the controller, to receive the first SMI, to handle the first SMI and to generate a wake-up signal after receiving the first SMI, wherein the wake-up signal references a first memory address of a default SMI handler; and

a second logical processor, coupled to the controller, to handle the first SMI after the wake-up signal is received from the first logical processor.

41. (Previously Amended) The apparatus of claim 40, wherein handling the first SMI with the first logical processor comprises executing the default SMI handler withthe first logical processor.

42. (Previously Amended) The apparatus of claim 41, wherein the first memory address is 1k aligned.

43. (Previously Amended) The apparatus of claim 41, wherein handling the first SMI with the second logical processor comprises executing the default SMI handler with the second logical processor.

44. (Original) The apparatus of claim 43, wherein handling the first SMI with the second logical processor further comprises patching an instruction pointer to a second memory address.

45. (Previously Amended) A system comprising:

a controller hub to generate a first system management interrupt (SMI);

a memory to hold code beginning at a first memory address;

a first processor coupled to the controller hub to receive the first SMI and to handle the first SMI, wherein the first processor is to execute the code beginning at the first memory address to handle the first SMI and is to generate a wake-up signal after receiving the first SMI; and

a second processor coupled to the controller hub to receive the first SMI, to be woken up in response to receiving the wake-up signal, and to handle the first SMI after receiving the wake-up signal, wherein the second processor is to execute the code beginning at the first memory address to handle the first SMI.

46. (Original) The system of claim 45, wherein the first and second processors are logical processors on the same die.

47. (Original) The system of claim 45, wherein the first and second processors are physical processors located on separate packages.

48. (Previously Amended) The system of claim 45, wherein a pin is toggled on the controller hub to generate the first SMI.

49. (Previously Amended) The system of claim 45, wherein boot code is executed by the controller hub to generate the first SMI.

50. (Previously Amended) The system of claim 45, wherein the code beginning at the first memory address includes SMI handling code.

51. (Previously Amended) The system of claim 50, wherein the wake-up signal includes a vector referencing the first memory address.

52. (Original) The system of claim 51, wherein handling the first SMI with the second processor after receiving the wake-up signal further comprises setting a pointer to a second memory address.

53. (Original) The system of claim 52, wherein the second processor, upon resuming from handling the SMI, executes code at the second memory address.

54. (Currently Amended) A system comprising:

a memory to hold system management interrupt (SMI) code beginning at a first memory address;

a first logical processor coupled to the memory to execute the SMI code beginning at the first memory address to handle an SMI for the first logical processor in response to receiving the SMI; and

a second logical processor coupled to the memory to execute the SMI code beginning at the first memory address to handle the SMI for the second logical processor in response to the SMI being software generated.

55. (Currently Amended) The system of claim 54, wherein the first logical processor is associated with a first system management base (SMBase) address.

56. (Currently Amended) The system of claim 55, wherein the second logical processor is associated with a second SMBase address.

57. (Previously Amended) The system of claim 56, wherein the first memory address has an offset from the first SMBase address.

58. (Previously Amended) The system of claim 57, wherein a target SMBase referenced by the SMI code, by default, is the first SMBase address.

59. (Currently Amended) The system of claim 58, wherein the target SMBase is changed to the second SMBase address before the second logical processor executes the SMI code.

60. (Cancelled)

61. (Cancelled)